

This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Currently Amended) A process for transferring pen data between unmanaged and managed code on a computing device for use by an application on the computing device, the unmanaged code being code native to and executed directly by a processor of the computing device, the managed code being executed in a common language run-time environment of a framework operating on the computing device, the common language run-time environment of the framework executing the managed code independent of a type of the processor of the computing device, the process comprising the steps of:

receiving pen data in a pen service operating in a user mode on the computing device and written in unmanaged code and separate from the application on the computing device, the pen data being generated by a digitizer of the computing device upon movement of a stylus with respect to a surface of the digitizer and being received by the pen service from a pen device driver operating in a kernel mode on the computing device, the pen data including at least one location on the digitizer of the stylus;

the pen service transferring information related to said pen data to a mutual exclusion shared memory on the computing device designated to be non-simultaneously shared between unmanaged code and managed code, and receiving from the shared memory a pointer to the information in the shared memory;

loading a pen input component into the application on the computing device, the pen input component being written in unmanaged code and being designated to communicate with the pen service for the application;

the pen service transferring the received pointer that points to said information in said shared memory to the loaded pen input component of the application;

the loaded pen input component further transferring the received pointer to a stylus input subsystem of the application on the computing device, the stylus input subsystem being separate from the pen input component and being written in managed code and executed in the common language run-time environment, the unmanaged code of the pen input component of the

application transferring the received pointer to the managed code of the stylus input subsystem of the application by way of a P-Invoke-style interface exposed therebetween;

the stylus input subsystem of the application receiving the transferred pointer from the pen input component of the application; and

the stylus input subsystem of the application submitting the received pointer with a retrieval command to the shared memory to retrieve said information from said shared memory by way of the transferred pointer, the retrieval command as submitted by the stylus input subsystem comprising a P-Invoke method "GetData" which takes into account a context of the stylus input subsystem, the retrieval command being submitted by way of the exposed P-Invoke-style interface to the pen input component;

the method further comprising the pen input component exposing the P-Invoke-style interface between the pen input component and the stylus input subsystem to cross between the managed and unmanaged code thereof.

2. (Original) The process according to claim 1, further comprising the steps of:
transferring additional information from said at least in part managed application to said shared memory;
transferring a pointer that points to said additional information to said component;
retrieving said additional information from said shared memory.

3. (Original) The process according to claim 1, further comprising the step of:
using a P-invoke style interface crossing between unmanaged code and managed code.

4. (Original) The process according to claim 1, further comprising the step of:
exchanging information through a COM interface.

5. (Original) The process according to claim 1, said component being a pen services component.

6. (Original) The process according to claim 1, said application including a pen input managed client.

7. (Original) The process according to claim 1, said component receiving input from at least one pen device driver.

8. (Currently Amended) A system for transferring information between unmanaged code and managed code on a computing device, the unmanaged code being code native to and executed directly by a processor of the computing device for use by an application on the computing device, the managed code being executed in a common language run-time environment of a framework operating on the computing device, the common language run-time environment of the framework executing the managed code independent of a type of the processor of the computing device, the system comprising:

a mutual exclusion shared memory on the computing device designated to be non-simultaneously shared between unmanaged code and managed code;

a pen service operating in a user mode on the computing device, the pen service being ~~that is~~ written in unmanaged code and ~~that receives~~ receiving pen data, the pen data being generated by a digitizer of the computing device upon movement of a stylus with respect to a surface of the digitizer and being received by the pen service from a pen device driver operating in a kernel mode on the computing device, the pen data including at least one location on the digitizer of the stylus, the pen ~~input component~~ service transferring information relating to said pen data to said shared memory[[,]] and receiving from the shared memory a pointer to the information in the shared memory,

a pen input component loaded into the application on the computing device, the pen input component being written in unmanaged code and being designated to communicate with the pen service for the application;

the pen service transferring the received pointer that points to said information in the shared memory to the loaded pen input component of the application;

the loaded pen input component further transferring the received pointer to a stylus input subsystem of the application on the computing device, the stylus input subsystem being separate from the pen input component and being managed code and executed in the common language run-time environment, the unmanaged code of the pen input component of the application transferring the received pointer to the managed code of the stylus input subsystem of the application by way of a P-Invoke-style interface exposed therebetween;

said stylus input subsystem of the application having managed code receiving the transferred pointer from the pen input component of the application and submitting the received pointer with a retrieval command to the shared memory to retrieve said information from said shared memory by way of the transferred pointer, the retrieval command as submitted by the stylus input subsystem comprising a P-Invoke method "GetData" which takes into account a context of the stylus input subsystem, the retrieval command being submitted by way of the exposed P-Invoke-style interface to the pen input component;

the pen input component exposing the P-Invoke-style interface between the pen input component and the stylus input subsystem to cross between the managed and unmanaged code thereof.

9. (Original) The system according to claim 8, said component exposing a COM interface.

10. (Original) The system according to claim 8, said application using a P-Invoke-style command.

11. (Original) The system according to claim 8, said component including a pen services component.

12. (Original) The system according to claim 8, further comprising:
at least one pen device driver sending information to said component.

13. (Original) The system according to claim 8, further comprising:
said application including a pen input managed client.

14. (Currently Amended) A computer-readable storage medium having a program stored thereon for transferring information related to ink between an unmanaged pen input component and a managed stylus input subsystem of an application on a computing device for use by an application on the computing device, the unmanaged pen input component being native to and executed directly by a processor of the computing device, the managed stylus input subsystem of the application being separate from the unmanaged pen input component and being executed in a common language run-time environment of a framework operating on the computing device, the common language run-time environment of the framework executing the managed code independent of a type of the processor of the computing device, said program comprising the steps of:

receiving pen data in a pen service operating in a user mode on the computing device and written in unmanaged code and separate from the application on the computing device, the pen data being generated by a digitizer of the computing device upon movement of a stylus with respect to a surface of the digitizer and being received by the pen service from a pen device driver operating in a kernel mode on the computing device, the pen data including at least one location on the digitizer of the stylus;

the pen service transferring information related to said pen data to a mutual exclusion shared memory on the computing device designated to be non-simultaneously shared between unmanaged code and managed code, and receiving from the shared memory a pointer to the information in the shared memory;

loading a pen input component into the application on the computing device, the pen input component being written in unmanaged code and being designated to communicate with the pen service for the application;

the pen service transferring the received pointer that points to said information in said shared memory to the loaded pen input component of the application;

the loaded pen input component further transferring the received pointer to a stylus input subsystem of the application on the computing device, the stylus input subsystem being separate from the pen input component and being written in managed code and executed in the common language run-time environment, the unmanaged code of the pen input component of the application transferring the received pointer to the managed code of the stylus input subsystem of the application by way of a P-Invoke-style interface exposed therebetween;

the stylus input subsystem of the application receiving the transferred pointer from the pen input component of the application; and

the stylus input subsystem of the application submitting the received pointer with a retrieval command to the shared memory to retrieve said information from said shared memory by way of the transferred pointer, the retrieval command as submitted by the stylus input subsystem comprising a P-Invoke method "GetData" which takes into account a context of the stylus input subsystem, the retrieval command being submitted by way of the exposed P-Invoke-style interface to the pen input component;

the pen input component exposing the P-Invoke-style interface between the pen input component and the stylus input subsystem to cross between the managed and unmanaged code thereof.

15. (Previously Presented) The computer-readable storage medium according to claim 14, said program further comprising the steps of:

transferring additional information from said at least in part managed application to said shared memory;

transferring a pointer that points to said additional information to said component;

retrieving said additional information from said shared memory.

16. (Previously Presented) The computer-readable storage medium according to claim 14, said program further comprising the step of:

using a P-invoke style interface crossing between unmanaged code and managed code.

17. (Previously Presented) The computer-readable storage medium according to claim 14, said program further comprising the step of:

exchanging information through a COM interface.

18. (Previously Presented) The computer-readable storage medium according to claim 14, said component being a pen services component.

19. (Previously Presented) The computer-readable storage medium according to claim 14, said application including a pen input managed client.

20. (Previously Presented) The computer-readable storage medium according to claim 14, said component receiving input from at least one pen device driver.